

# A Laser Plotting System for VLSI Chip Layouts

J. A. Harding and L. J. Deutsch  
Communications Systems Research Section

*One of the most time consuming facets of custom Very Large Scale Integration (VLSI) design is obtaining hardcopy plots of the mask geometries of cells and chips. The traditional method of generating these plots is to use a multicolor pen plotter. Pen plotters are inherently slow and the plotting speed increases linearly with the number of edges that must be plotted. A moderate custom chip design at JPL now consists of more than 200,000 such edges and can take as much as eight hours to plot using a pen plotter. This paper describes software that was written at JPL to produce similar plots using a laser printer. It is shown that, for rather small layouts, the laser printer can provide nearly instantaneous turnaround. For moderate to large chip designs, the laser printer provides a factor of five or more improvement in speed over pen plotting.*

## I. Introduction

The design of a custom or semicustom Very Large Scale Integrated (VLSI) chip involves many Computer Aided Design (CAD) tools. A UNIX-based CAD system for VLSI design was described in Ref. 1. The process is centered around the creation of a computer file that describes the geometry for each mask layer in the fabrication process. This data is often referred to as a *layout*. Caltech Intermediate Format (CIF) (Ref. 2) is simply a standardized syntax for the description of such geometry.

There are several stages in the design process when it is preferable that the designer have a hardcopy plot of a layout file. One of these stages is called floorplanning. In floorplanning, layouts of subcells are arranged into a larger system. This involves a great deal of rearranging of the subcells. Hardcopy plots help the designer understand the relative sizes, interconnection requirements, and shapes of the subcells.

Hardcopy plots are also useful in the chip simulation process. If the plots have node labels on them, then the designer can use them to follow a circuit's behavior during simulation.

Plots are sometimes helpful even while performing layout editing – even when using a graphical layout editor such as *caesar* (Ref. 3). These plots help the designer see areas of the chip that are too large to be viewed on the graphics screen at one time.

All of these applications of hardcopy *cifplots* are only feasible if the plots can be generated in a relatively short time. Currently, *cifplots* are generated on mechanical pen plotters. These devices are inherently slow and are only useful in the design process for very small cells. The turnaround time for generating pen plots is, as a rule, only acceptable for layouts of cells with up to around 50 transistors. Larger cells require more time to plot than is practical for applications other than producing documents.

The time that it takes a pen plotter to produce a cifplot is roughly proportional to the number of lines that it must plot multiplied by the length of these lines (this is the arclength of the CIF file). Most pen plotting programs for cifplotting use a different color pen for each mask layer. This has definite advantages. However, only the outline of each geometric shape is plotted. This means that if the plot is even moderately dense, it is often difficult to tell where the interior and exterior regions of these shapes really are. Many pen plotting programs are not capable of placing text on the plot. This means that electrical nodes cannot be labeled and the plot will be of limited use during the chip simulation process. Finally, most existing software for cifplotting — regardless of the output device — is not technology independent. This means that the programs are limited to a small number of fabrication technologies and therefore they will always become obsolete.

With the complexity of JPL in-house chip designs increasing by about a factor of five every year, it became necessary this year to address the hardcopy plotting issue. In an attempt to solve most of the above problems, a laser printer was purchased. The particular printer is a Quality Micro Systems (QMS) Lasergrafix 2400. The QMS printer is capable of producing high quality text at a rate of 24 pages each minute. It also comes with special hardware and firmware for performing many complex graphics functions. It may be operated in a pure vector mode similar to that of a pen plotter only many times faster. The resolution of the QMS printer is 300 points per inch in both axes. It uses 8.5 in. X 11 in. plain paper.

A laser printer is a device that uses a plain paper copying process to produce rasterized hardcopy output. In the case of the QMS printer, the method is the Xerographic copying process. In fact, the QMS is based on a standard Xerox brand copier. A laser is used to scan input pixel information onto a drum. A sheet of plain paper is passed over the drum and the pixels that will be black are heated with respect to the rest of the paper. A dry toner is then applied to the paper. It fixes to the paper only where the paper has been heated by the drum. The remaining toner is then removed resulting in a finished page.

Software was written to produce cifplots on the QMS printer. The software is called *Laser*. Plots generated by *Laser* are produced in black and white. This is a disadvantage over the pen plotters. However, the geometric shapes may be filled in with different hatching styles. This gives the same functionality as the different colors, and it also solves the problem of distinguishing interior and exterior regions in complex chip plots. *Laser* is completely technology independent. The user may use a technology description file in a standard system library or one of original design. By using the

QMS printer's text capability, *Laser* can also label electrical nodes. Finally, *Laser* works much faster than programs that use pen plotters. It is now possible to obtain hardcopy cifplots of chips with hundreds of transistors in a matter of minutes. In fact, since the time required to produce *Laser* plots is roughly proportional to the number of geometric shapes rather than arc length or number of edges, the *Laser* printer has a distinct speed advantage over pen plotters.

Since all the designs that are produced on the UNIX-based CAD system are of *Manhattan* geometry (i.e., they consist of rectangles that are all aligned with the *x* and *y* axes) the ability to plot non-*Manhattan* CIF was not programmed into *Laser*. Only a minor modification would be required to allow *Laser* to plot non-*Manhattan* (sometimes called *Boston*) geometry.

## II. The Laser Cifplotting Software

*Laser* actually consists of several individual programs that are controlled through a command file (called a *shell script* in UNIX). The command file is written in the UNIX C-shell. It parses the command line and routes the appropriate data through the various programs. The user invokes *Laser* by typing the word *Laser*. In addition, the user may specify certain options on the command line.

There are currently three options supported on the command line. First, a technology description file may be specified. The default technology is currently NMOS. The *Laser* command file first searches the system's technology library for the requested file. If it is not found there, then the user's current directory is searched. In this way, each user may have a set of unique technology description files. If a designer were working with an NMOS design, for example, but wished to use a hatching style for the layers that is different than the one in the system library, then a technology description file with a name such as *nmos.my-own* could be created and placed in the user's own directory.

Another option allows the user to specify which layers should be filled in and which should be plotted as outlines only. This option is provided to allow even quicker turnaround time for simple cell layouts. Since node labels are treated like a mask layer in the *Laser* program, the user could use this option to eliminate the plotting of labels. This might be necessary on very complex plots in which the labels would overlap each other. *Laser* creates a legend on the finished plot that indicates the hatch styles that are used for each non-deleted layer.

The last option that the user may request on the command line is the *title* option. The user may use this option to pass a

line of text to the Laser program for use as a large headline-style title that will be placed at the top of the plot.

The first program that is called by the Laser command file is a University of California at Berkeley program called *cifflatten*. CIF is a hierarchical description language. This means that cells that are used many times in a chip design need only be described once at a geometric level. Further references to that cell may then be specified by calls to this definition. *Cifflatten* takes a CIF file as an input and produces a CIF file that describes the same design but with all the hierarchy removed. This "simple" CIF is easier to process in subsequent parts of the program. *Cifflatten* was slightly modified to allow the calculation of the overall geometric dimensions of the chip design at the same time as the flattening operation. This information is needed to scale the plot for final output on a hardcopy page.

After flattening, the CIF information is processed by a program called *qmscif*. *Qmscif* is a Pascal program that was written here at JPL. It produces a file that contains the control sequences for the QMS printer that are needed to produce the plot. QMS calls this set of codes the *QUIC* programming language. The *QUIC* language has graphics primitives for vector plotting, box drawing, and labeling. It even has a simple looping ability that is used in *qmscif* to produce the various hatch styles for different VLSI layers.

The *QUIC* file is placed in the UNIX spooling system for subsequent downloading to the QMS printer. Laser is fully integrated onto the UNIX spooling environment. This means that *cifplotting*, laser text printing, and standard (dot matrix) text printing are all under one set of system controls.

When the *QUIC* file reaches the top of the spooler's queue, it is sent to the QMS printer. A plot is then generated.

### III. Some Benchmark Tests

In order to determine the functionality and relative performance improvement over pen plotting, a set of CIF files that covers a wide range of typical designs was assembled. Many of these CIF files come from designs that were performed for projects within the DSN Advanced Systems Program. Table 1 shows some of the important statistics of these designs.

The names of the designs are shown in the first column of the table. *Shiftcell* is a single half-bit stage in a dynamic NMOS shift register. It is the simplest of the designs that were tested. *Fa* is a full adder. It is typical of the complexity of a single low-level subcell in a chip design. *D* is a d-type flip-flop. It comes from a library of standard cells that are

based on the Sandia National Laboratory radiation-hardened CMOS fabrication process. *Lag* is a bit slice through a three-level complex correlator with phase rotation. It is typical of an intermediate level subcell, and it contains several cells of the complexity of *fa*. *Conv2* is a small complete chip. It is the layout of the Multicode Convolutional Encoder chip (Ref. 4). *Rsint* is the layout of an 8-bit (255, 223) Reed-Solomon encoder with interleaving (Ref. 5). It is an example of a chip of moderate complexity. No chips of higher complexity than *rsint* were benchmarked because the time for pen plotting would have been prohibitive.

The third column of the table shows the number of transistors in each design. It gives a good indication of the relative complexity of the designs. It is important to note that the increase in complexity between *conv2* and *rsint* is typical of the trend over the two-year time period that this represents here at JPL. A typical large chip that is designed in-house today contains between 30,000 and 60,000 transistors.

The fourth column indicates the number of rectangles in the design. Since all these designs are of Manhattan geometry, this is actually the number of individual shapes in the design. The Laser plotting speed should be roughly proportional to this number.

The next column is labeled *Normalized Arc Length*. This number, when divided by 1,000, is actually the number of millimeters of perimeter for all the rectangles that would be measured on the fabricated chip using a standard fabrication scaling ( $\lambda = 2.0\mu$ ). The time for pen plotting should be proportional to this number. Notice that ratio of the largest to the smallest number of rectangles is about 3,200 while the similar ratio in arc length is about 4,300. This already indicates a probable speed advantage for Laser plotting, even if the time to plot a shape is the same on the two devices.

The last column in the table indicates the *regularity ratio*. This is a measure of the hierarchical complexity of the design. It is actually the number of rectangles in the design divided by the number of rectangles defined explicitly (rather than by calls to cell definitions) in the CIF file. A value of 1.0 means that there is no hierarchy in the CIF. The higher the ratio, the more regular the design is. *Rsint* is a very regular design that contains large blocks of identical cells. The regularity ratio is an indication of the amount of computation needed to perform the CIF flattening operation.

These designs were plotted using both the pen plotter and the QMS Laser printer. The plots were performed on a Digital Equipment Corporation VAX 11/750 computer with a load average of approximately 3.0 (moderately heavy usage). The plotting and processing times (actual elapsed time) were both

recorded. The processing time for the pen plotter is not important since it is negligible compared to the mechanical plotting time. Also, since the pen plotter is not a spooled device, it is run directly by the software and the processing time cannot be distinguished.

The results of these timing tests are recorded in Table 2. The pen plots of the six CIF files appear in Figs. 1 through 6. The Laser plots for the same files appear in Figs. 7 through 12.

#### IV. Conclusions

The first thing that is evident from the figures is that the Laser plots are much easier to read. This is because the QMS printer fills in the various layers. The layer types and intersections of layers are all easily discernible. Even though the pen plots were originally in color, the color effect is much less pleasing to the eye than the solid regions produced by the Laser printer.

Table 2 shows the processor and plotting times required for both the pen and Laser plots. Notice that in every case, the Laser plot took less time to generate than the corresponding pen plot. Even in the smallest CIF file (shiftcell) there was an almost three-fold increase in performance. In fact, the ratio

of Laser plotting time to pen plotting time increases with the complexity of the CIF file as expected.

#### V. Future Work

Some improvements will be made in Laser during the next year. One of these will be to add the ability to produce mosaic plots. Since the QMS printer is limited to a paper size of 8.5 in.  $\times$  11 in., large CIF files for which the designer wishes to see a reasonable amount of detail will have to be generated on multiple pages. These could then be pasted together to produce one large plot. This technique is also used with the current pen plotter which is limited to 11 in.  $\times$  14 in. paper.

In order to make the software run faster, the code in qmscif should be merged with ciffatten so that multiple interrogations of very large data files need not occur during processing. Another advantage of doing this would be to allow some cells of a CIF file to have only their outlines plotted.

As costs become less prohibitive, a color Laser printer could be acquired. This would enhance the plots even more and allow many more layer types to be plotted on a page without confusion.

#### References

1. Deutsch, L. J., "An Integrated UNIX-Based CAD System for the Design and Testing of Custom VLSI Chips," *TDA Progress Report 42-81*, Jet Propulsion Laboratory, Pasadena, California, January-March 1985, pp. 51-62.
2. Mead, C. and Conway, L., *Introduction to VLSI Systems*, Addison-Wesley Publishing Company, Menlo Park, California, 1980, pp. 115-127.
3. Ousterhout, J., *Editing VLSI Circuits with Caesar*, Computer Science Division, University of California, Berkeley, California, 1983.
4. Deutsch, L. J., "A VLSI Implementation of a Multicode Convolutional Encoder," *TDA Progress Report 42-72*, Jet Propulsion Laboratory, Pasadena, California, February 15, 1983, pp. 61-69.
5. Truong, T. K., Deutsch, L. J., Reed, I. S., Hsu, I. S., Wang, K., Yeh, C. S., "The VLSI Design of a Reed-Solomon Encoder Using Berlekamp's Bit Serial Algorithm," *Proceedings of the Third Caltech Conference on VLSI*, California Institute of Technology, Pasadena, California, 1983, pp. 303-330.

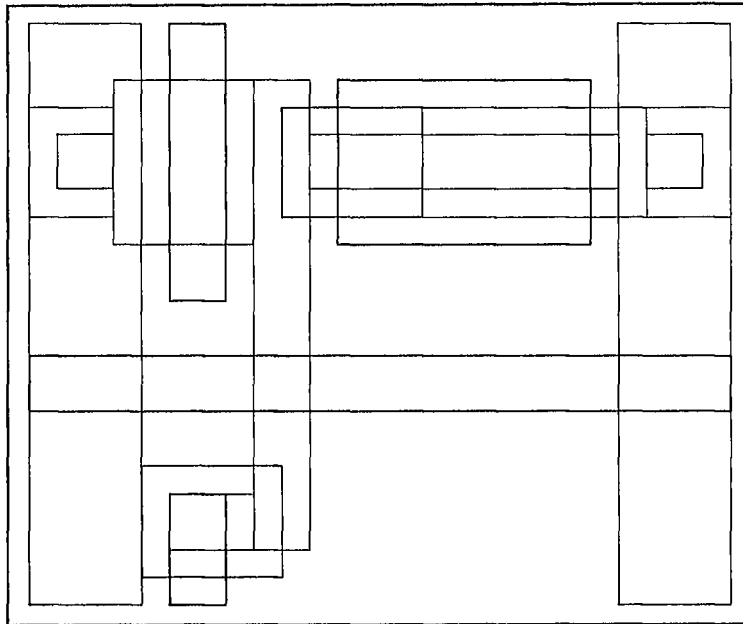
**Table 1. Statistics for benchmark test layouts**

Cell Name	Fabrication Technology	Number of Transistors	Number of Rectangles	Normalized Arc Length	Regularity Ratio
shiftcell	NMOS	31	17	324	1.0
fa	NMOS	18	114	2,378	1.0
d	CMOS	25	288	11,364	1.0
lag	NMOS	184	1676	44,956	2.6
conv2	NMOS	453	5646	186,632	4.0
rsint	NMOS	8918	54182	1,405,182	27.1

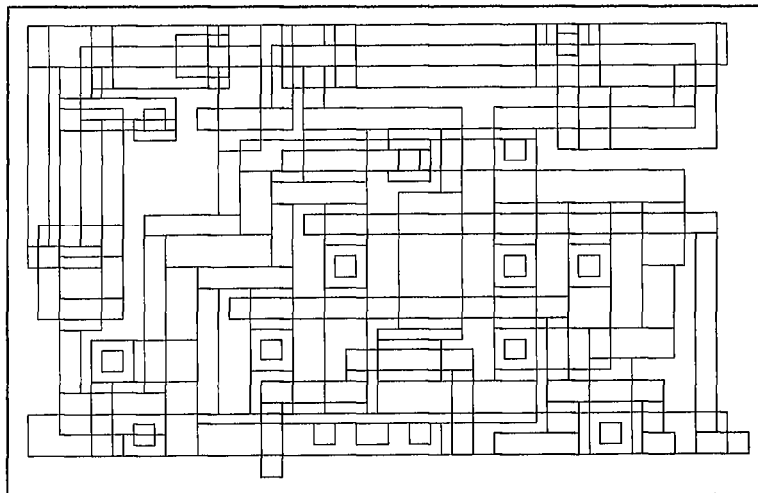
**Table 2. Results of benchmark timing tests for pen and laser CIF plotting**

Cell Name	Time for CIF Processing	Time to Download to Laser Printer	Total Time for Laser Plotting	Time for Pen Improvement	Relative Speed
shiftcell	0:13	0:16	0:39	1:05	2.7
fa	0:23	0:38	1:01	3:46	3.7
d	0:46	1:12	1:58	6:58	3.5
lag	1:48	1:28	2:36	13:24	5.2
conv2	11:10	3:38	14:48	42:40	2.9
rsint	37.40	26.36	1:03:56	5:51:00	5.5

Note: All times are in the form *hours:minutes:seconds*



**Fig. 1. Pen cifplot of *shiftcell***



**Fig. 2. Pen cifplot of *fa***

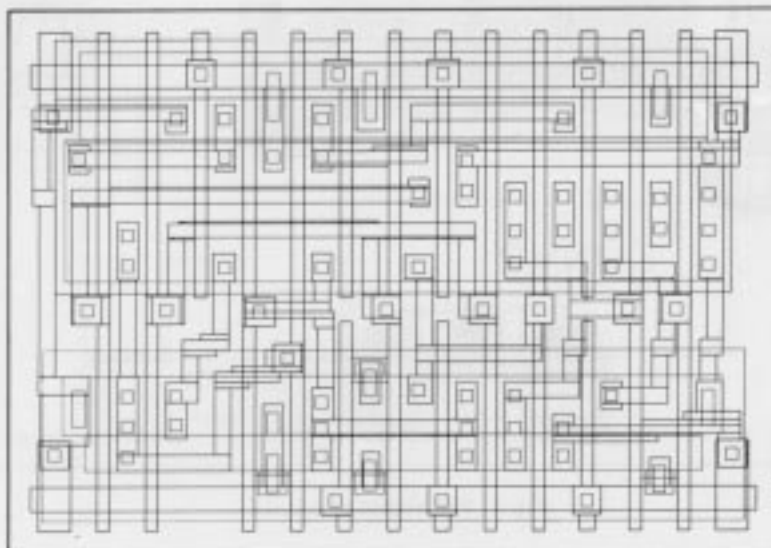


Fig. 3. Pen cilplot of  $d$

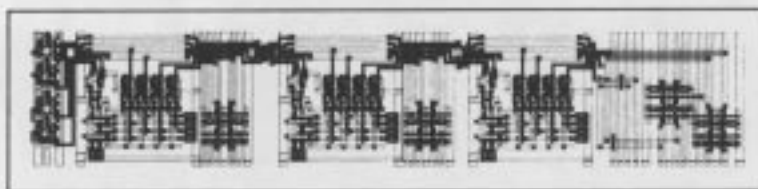


Fig. 4. Pen cilplot of  $lag$

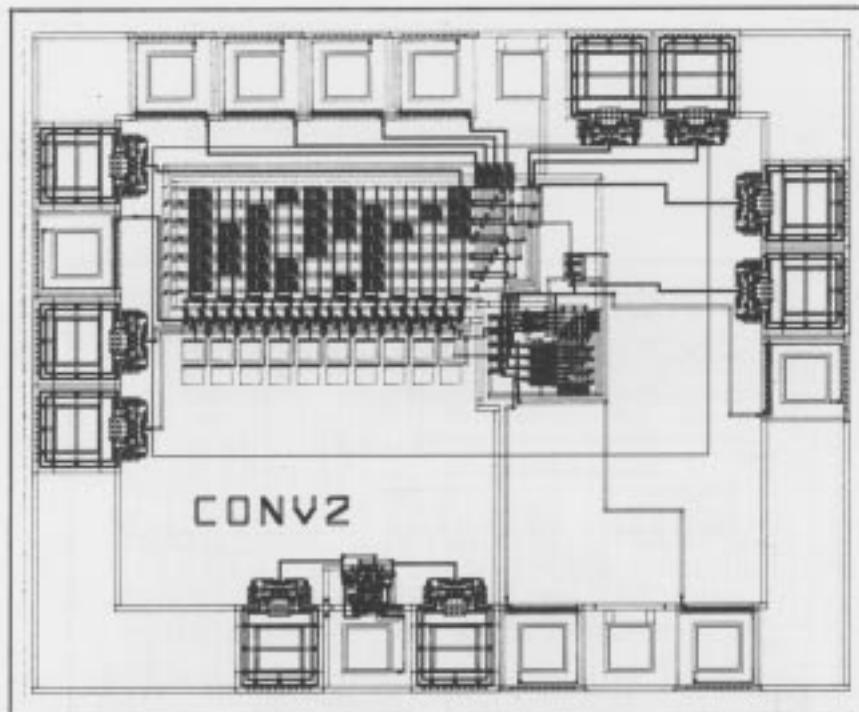


Fig. 5. Pen cilplot of conv2

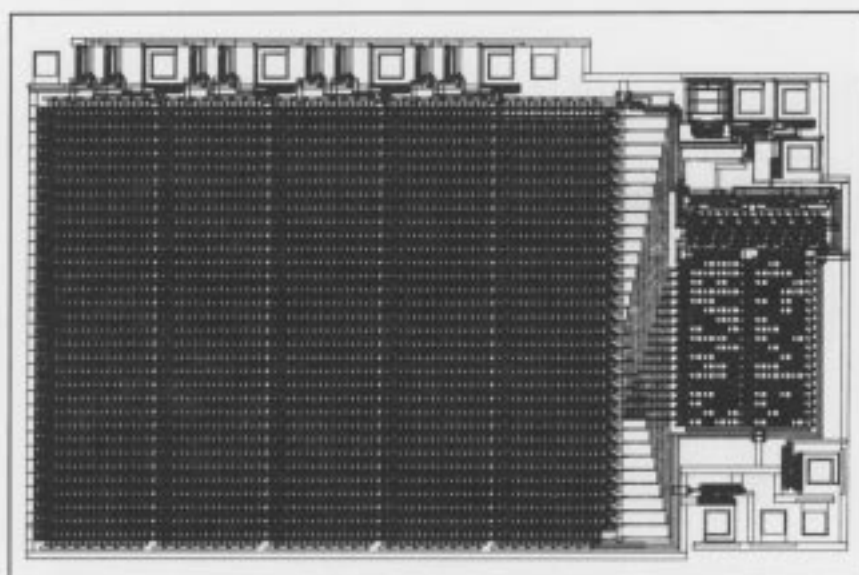


Fig. 6. Pen cilplot of rsint



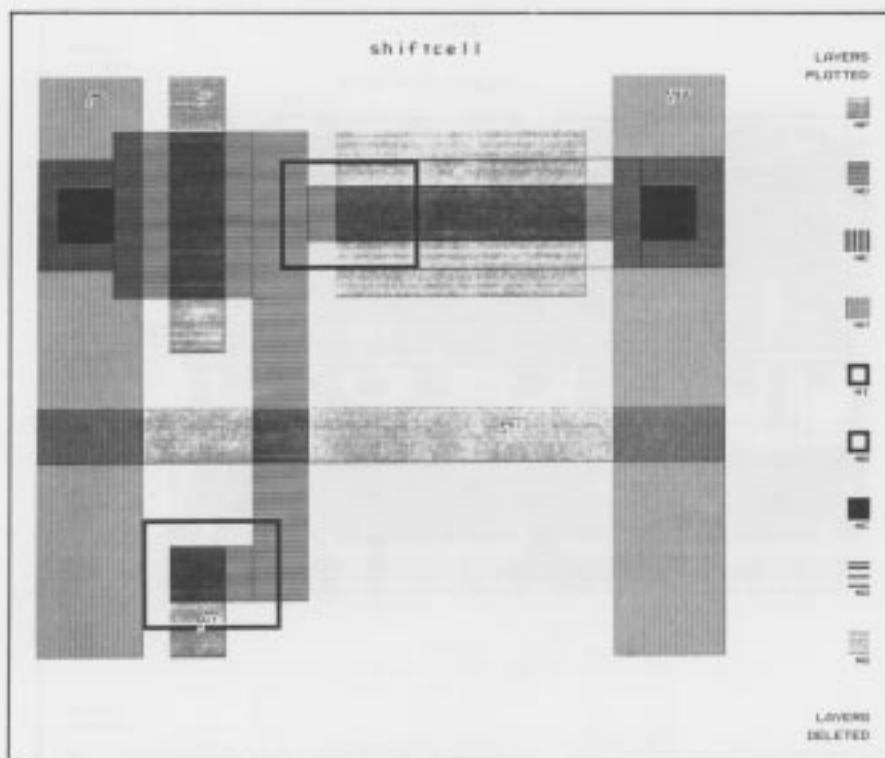


Fig. 7. Laser citplot of *shiftcell*

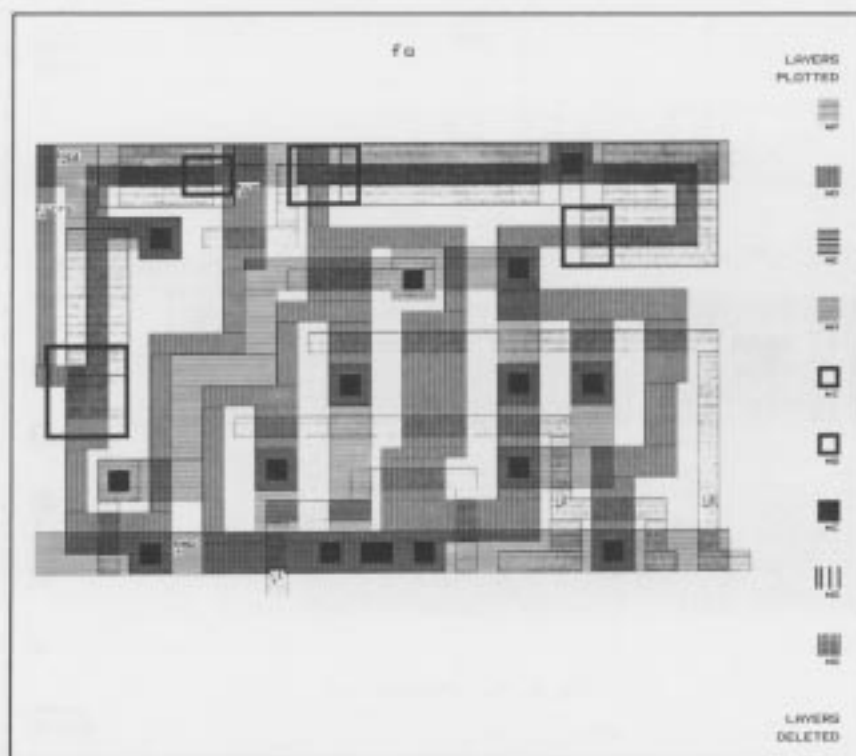


Fig. 8. Laser citplot of *fa*

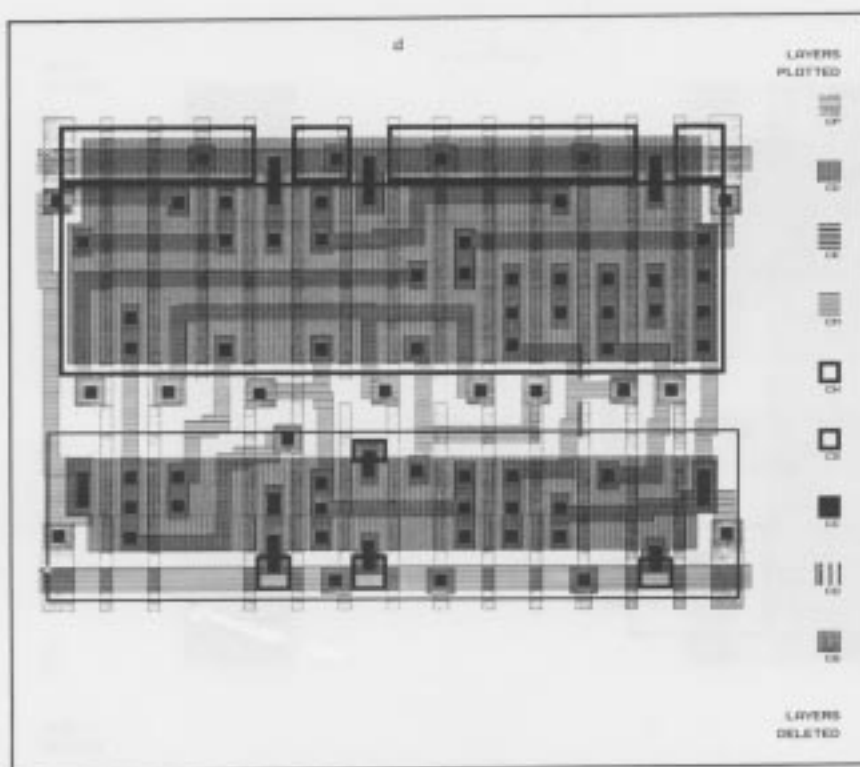


Fig. 9. Laser citplot of *d*

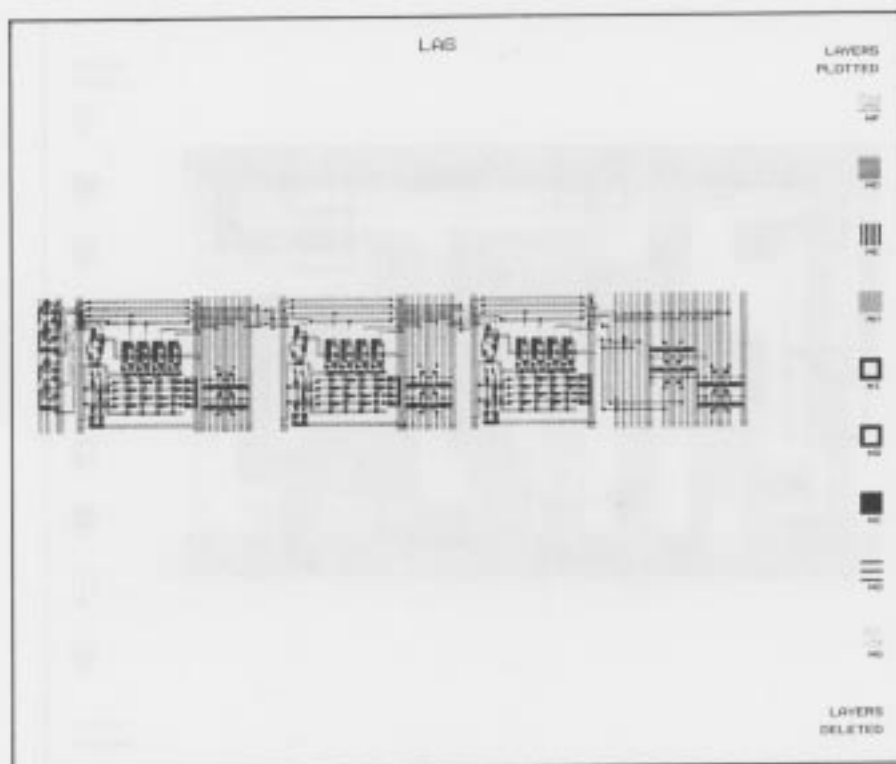


Fig. 10. Laser citplot of *lag*

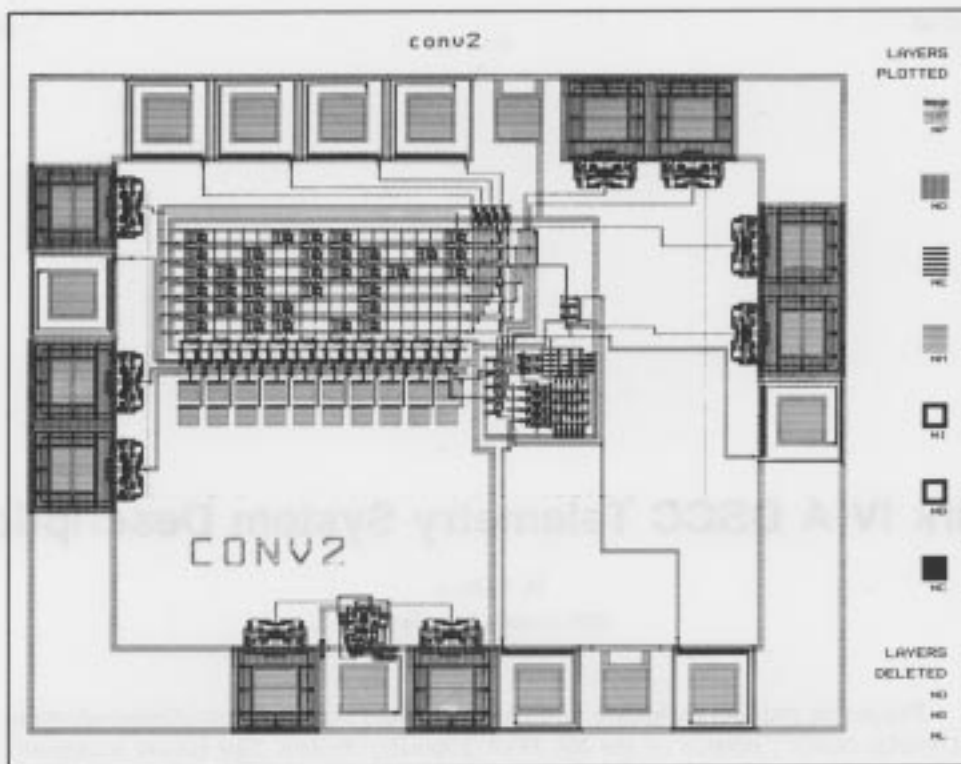


Fig. 11. Laser ctfplot of conv2

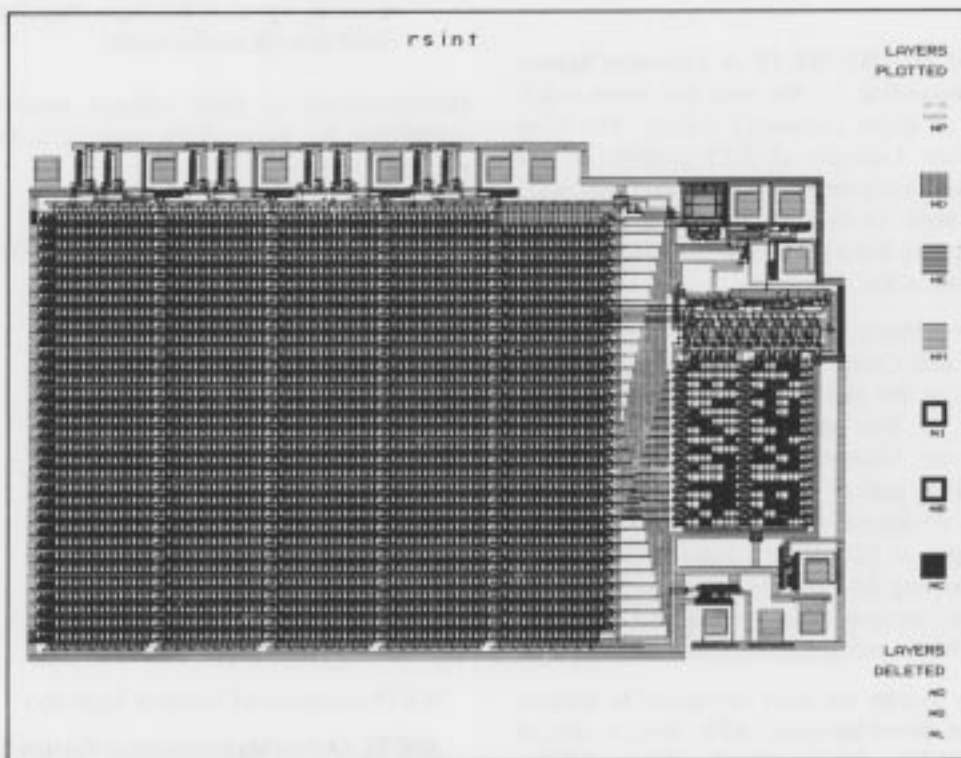


Fig. 12. Laser ctfplot of rsint